WHAT IS CLAIMED IS:

1. A system for providing a floating point sum, comprising:

an analyzer circuit configured to determine a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

a results circuit coupled to the analyzer circuit and configured to assert a resulting floating point operand containing the sum of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand.

2. The system for providing a floating point sum of claim 1, wherein the analyzer circuit further comprises:

a first operand buffer configured to store the first floating point operand;

a second operand buffer configured to store the second floating point operand;

a first operand analysis circuit coupled to the first operand buffer, the first operand analysis circuit configured to generate a first characteristic signal having information relating to the first status; and

a second operand analysis circuit coupled to the second operand buffer, the second operand analysis circuit configured to generate a second characteristic signal having information relating to the second status.

3. The system for providing a floating point sum of claim 2, wherein the first status and the second status are determined without regard to memory storage external to the first operand buffer and the second operand buffer.

4. The system for providing a floating point sum of claim 3, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

5. The system for providing a floating point sum of claim 1, wherein the results circuit further comprises:

an adder circuit coupled to the analyzer circuit, the adder circuit configured to produce the sum of the first floating point operand and the second floating point operand;

an adder logic circuit coupled to the analyzer circuit and configured to produce the resulting status based upon the first status and the second status; and

a result assembler coupled to the adder circuit and the adder logic circuit, the result assembler configured to assert the resulting floating point operand and embed the resulting status within the resulting floating point operand.

6. The system for providing a floating point sum of claim 5, wherein the adder logic circuit is organized according to the structure of a decision table.

7. The system for providing a floating point sum of claim 1, wherein the sum of the first floating point operand and the second floating point operand is identical in all cases to the sum that would be produced if the two operands were first swapped.

8. The system for providing a floating point sum of claim 1, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

9. The system for providing a floating point sum of claim 8, wherein the overflow status represents one in a group of a +OV status and a -OV status.

10. The system for providing a floating point sum of claim 8, wherein the overflow status is represented as a predetermined non-infinity numerical value.

11. The system for providing a floating point sum of claim 8, wherein the underflow status represents one in a group of a +UN status and a -UN status.

12. The system for providing a floating point sum of claim 8, wherein the underflow status is represented as a predetermined non-zero numerical value.

13. The system for providing a floating point sum of claim 8, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

14. The system for providing a floating point sum of claim 8, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

15. A method for providing a floating point sum, comprising:

determining a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

asserting a resulting floating point operand containing the sum of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand.

16. The method for providing a floating point sum of claim 15, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer;

storing the second floating point operand in a second operand buffer;

generating a first characteristic signal representative of the first status; and

generating a second characteristics signal representative the second status.

17. The method for providing a floating point sum of claim 16, wherein the first characteristic signal and the second characteristic signal are generated without regard to memory storage external to the first operand buffer and the second operand buffer.

18. The method for providing a floating point sum of claim 17, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

19. The method for providing a floating point sum of claim 15, wherein the asserting stage further comprises:

producing the sum of the first floating point operand and the second floating point operand;

producing the resulting status based upon the first status and the second status; and

asserting the resulting floating point operand.

20. The method for providing a floating point sum of claim 15, wherein the sum of the first floating point operand and the second floating point operand is identical in all cases to the sum that would be produced if the two operands were first swapped.

21. The method for providing a floating point sum of claim 15, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

22. The method for providing a floating point sum of claim 21, wherein the overflow status represents one in a group of a +OV status and a -OV status.

23. The method for providing a floating point sum of claim 21, wherein the overflow status is represented as a predetermined non-infinity numerical value.

24. The method for providing a floating point sum of claim 21, wherein the underflow status represents one in a group of a +UN status and a -UN status.

25. The method for providing a floating point sum of claim 21, wherein the underflow status is represented as a predetermined non-zero numerical value.

26. The method for providing a floating point sum of claim 21, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

27. The method for providing a floating point sum of claim 21, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

28. A computer-readable medium on which is stored a set of instructions for providing a floating point sum, which when executed perform stages comprising:

determining a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

asserting a resulting floating point operand containing the sum of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand.

29. The computer-readable medium of claim 28, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer;

storing the second floating point operand in a second operand buffer;

generating a first characteristic signal representative of the first status; and

generating a second characteristic signal representative of the second status.

30. The computer-readable medium of claim 29, wherein the first characteristic signal and the second characteristic signal are generated without regard to memory storage external to the first operand buffer and the second operand buffer.

31. The computer-readable medium of claim 30, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

32. The computer-readable medium of claim 28, wherein the asserting stage further comprises:

producing the sum of the first floating point operand and the second floating point operand;

producing the resulting status based upon the first status and the second status; and

asserting the resulting floating point operand.

33. The computer-readable medium of claim 28, wherein the sum of the first floating point operand and the second floating point operand is identical in all cases to the sum that would be produced if the two operands were first swapped.

34. The computer-readable medium of claim 28, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

35. The computer-readable medium of claim 34, wherein the overflow status represents one in a group of a +OV status and a -OV status.

36. The computer-readable medium of claim 35, wherein the overflow status is represented as a predetermined non-infinity numerical value.

37. The computer-readable medium of claim 34, wherein the underflow status represents one in a group of a +UN status and a -UN status.

38. The computer-readable medium of claim 37, wherein the underflow status is represented as a predetermined non-zero numerical value.

39. The computer-readable medium of claim 34, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

40. The computer-readable medium of claim 34, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.